

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日 2 0 0 3 年 4 月 1 4 日
Date of Application:

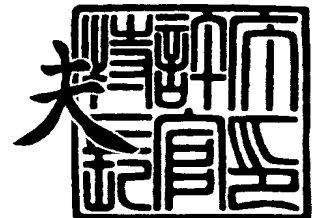
出 願 番 号 特 願 2 0 0 3 - 1 0 9 0 5 3
Application Number:
[ST. 10/C] : [J P 2 0 0 3 - 1 0 9 0 5 3]

出 願 人 ソニー株式会社
Applicant(s):

2 0 0 4 年 3 月 2 日

特許庁長官
Commissioner,
Japan Patent Office

今 井 康 夫



【書類名】 特許願

【整理番号】 0390053202

【提出日】 平成15年 4月14日

【あて先】 特許庁長官 太田 信一郎 殿

【国際特許分類】 H04N 7/26

【発明者】

 【住所又は居所】 東京都品川区北品川 6 丁目 7 番 3 5 号ソニー株式会社内

 【氏名】 富樫 治夫

【特許出願人】

 【識別番号】 000002185

 【氏名又は名称】 ソニー株式会社

【代理人】

 【識別番号】 100082740

 【弁理士】

 【氏名又は名称】 田辺 恵基

【手数料の表示】

 【予納台帳番号】 048253

 【納付金額】 21,000円

【提出物件の目録】

 【物件名】 明細書 1

 【物件名】 図面 1

 【物件名】 要約書 1

 【包括委任状番号】 9709125

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 符号化装置及び符号化方法

【特許請求の範囲】

【請求項 1】

入力された画像データをフィルタリング処理を施すことによりフィルタリング係数を生成するフィルタリング生成手段と、

上記フィルタリング係数を各画素の最上位ビットから最下位ビットまでの複数のビットプレーンに分割する分割手段と、

上記複数のビットプレーンのうち、下位側から所定数の上記ビットプレーンを取り除いた後、残りの各上記ビットプレーンのみを読み出して並列的に出力する読出制御手段と、

上記読出制御手段により並列的に出力された上記複数のビットプレーンについてそれぞれ符号化処理する複数の符号化手段と

を具え、各上記符号化手段が上記符号化処理を行う際に 1 フレーム当りの発生符号量を一定にさせるように、上記読出制御手段は、取り除く上記ビットプレーンの数を決定する

ことを特徴とする符号化装置。

【請求項 2】

上記読出制御手段は、

上記記憶手段に記憶されている各上記ビットプレーンのうち下位側かつ上記階層レベルの低い側から所定数の上記ビットプレーンを取り除く

ことを特徴とする請求項 1 に記載の符号化装置。

【請求項 3】

各上記符号化手段による符号化処理の結果に基づいて、1 フレーム当りの発生符号量を一定にさせるためにフィードフォワードさせるレート制御手段

を具えることを特徴とする請求項 1 に記載の符号化装置。

【請求項 4】

入力された画像データをフィルタリング処理を施すことによりフィルタリング係数を生成する第 1 のステップと、

上記フィルタリング係数を各画素の最上位ビットから最下位ビットまでの複数のビットプレーンに分割する第 2 のステップと、

分割された上記複数のビットプレーンのうち、下位側から所定数の上記ビットプレーンを取り除いた後、残りの上記複数のビットプレーンのみを読み出して並列的に出力する第 3 のステップと、

並列的に出力された上記複数のビットプレーンについてそれぞれ符号化処理する第 4 のステップと

を具え、上記第 4 のステップにおいて各上記符号化処理を行う際に 1 フレーム当りの発生符号量を一定にさせるように、上記第 3 のステップでは、取り除く上記ビットプレーンの数を決定する

ことを特徴とする符号化方法。

【請求項 5】

上記第 3 のステップでは、

各上記ビットプレーンのうち下位側かつ上記階層レベルの低い側から所定数の上記ビットプレーンを取り除く

ことを特徴とする請求項 4 に記載の符号化方法。

【請求項 6】

上記第 4 のステップにおける各上記符号化処理の結果に基づいて、1 フレーム当りの発生符号量を一定にさせるために必要な上記第 3 のステップにおいて取り除く上記ビットプレーンの数を設定した後、当該第 3 のステップにフィードバックさせる第 6 のステップ

を具えることを特徴とする請求項 4 に記載の符号化方法。

【発明の詳細な説明】

【0 0 0 1】

【発明の属する技術分野】

本発明は符号化装置及び符号化方法に関し、例えば J P E G (Joint Photographic Experts Groupe) 2 0 0 0 規格に準拠した符号化装置及び符号化方法に適用して好適なものである。

【0 0 0 2】

【従来の技術】

近年、新しいデータ圧縮方式として、J P E G 2 0 0 0 規格と呼ばれる圧縮方式が規格化されている。

【0 0 0 3】**【発明が解決しようとする課題】**

ところで、かかる J P E G 2 0 0 0 規格による符号化方式では、符号化処理が煩雑で伝送時間の短縮化を図ることが望まれていた。

【0 0 0 4】

本発明は以上の点を考慮してなされたもので、符号化処理に要する伝送効率を格段と向上し得る符号化装置及び符号化方法を提案するものである。

【0 0 0 5】**【課題を解決するための手段】**

かかる課題を解決するため本発明においては、入力された画像データをフィルタリング処理を施すことによりフィルタリング係数を生成するフィルタリング生成手段と、フィルタリング係数を各画素の最上位ビットから最下位ビットまでの複数のビットプレーンに分割する分割手段と、複数のビットプレーンのうち、下位側から所定数のビットプレーンを取り除いた後、残りの各ビットプレーンのみを読み出して並列的に出力する読出制御手段と、読出制御手段により並列的に出力された複数のビットプレーンについてそれぞれ符号化処理する複数の符号化手段とを設け、各符号化手段が符号化処理を行う際に 1 フレーム当りの発生符号量を一定にさせるように、読出制御手段は、取り除くビットプレーンの数を決定するようにした。

【0 0 0 6】

この結果この符号化装置では、ユーザから見て比較的細かい情報である部分に相当する所定数分のビットプレーンを取り除いた分だけ、各符号化手段がそれぞれ符号化処理を行う前段階での処理時間の変動を未然に防止することができる。

【0 0 0 7】

また本発明においては、入力された画像データをフィルタリング処理を施すことによりフィルタリング係数を生成する第 1 のステップと、フィルタリング係数

を各画素の最上位ビットから最下位ビットまでの複数のビットプレーンに分割する第2のステップと、分割された複数のビットプレーンのうち、下位側から所定数のビットプレーンを取り除いた後、残りの複数のビットプレーンのみを読み出して並列的に出力する第3のステップと、並列的に出力された複数のビットプレーンについてそれぞれ符号化処理する第4のステップとを設け、第4のステップにおいて各符号化処理を行う際に1フレーム当りの発生符号量を一定にさせるように、第3のステップでは、取り除くビットプレーンの数を決定するようにした。

【0008】

この結果この符号化方法では、ユーザから見て比較的細かい情報である部分に相当する所定数分のビットプレーンを取り除いた分だけ、各符号化処理を行う前段階での処理時間の変動を未然に防止することができる。

【0009】

【発明の実施の形態】

以下図面について、本発明の一実施の形態を詳述する。

【0010】

(1) 本実施の形態による符号化装置の構成

図1において、1は全体として符号化装置を示し、外部からビデオ入力部(VIN: Video In) 2に動画像データ(例えばITU-R BT656規格に対応する標準動画像信号) D1が入力されると、このビデオ入力部2は、CPU3の制御の下で、当該動画像データに時間多重されている各コンポーネントデータ(輝度Y及び色差Cb、Cr)を分離した後、当該各コンポーネントデータD2ごとに画像符号化領域を抽出してレジスタR1Eに送出する。

【0011】

レジスタR1Eは、ビデオ入力部2から与えられた各コンポーネントデータD2を一旦格納すると共に、それぞれ元の画像データの1フレーム分ごとに分けて順次インターフェースPSGを介してSDRAM(Synchronous Dynamic Random Access Memory) 4に記憶させる。その際、レジスタR1Eは、各コンポーネントデータD2ごとに1フレーム内のSDRAM4のアドレスを発生するようにな

されている。

【0012】

続いてCPU 3は、SDRAM 4に記憶されたフレーム単位の各コンポーネントデータD 2を読み出してインターフェースPSG及びレジスタR 2Eを介してウェーブレット変換部（DWT：Discrete Wavelet Transform）5に送出する。

【0013】

その際レジスタR 2Eでは、後述するウェーブレット変換に必要な画像の順番情報を管理しており、各コンポーネントデータD 2ごとに1フレーム内のSDRAM 4のアドレスを発生する。

【0014】

このウェーブレット変換部5は、低域周波数通過型及び高域周波数通過型の2分割フィルタ（図示せず）を有し、当該各2分割フィルタを用いて、各コンポーネントデータD 2に基づく画像をそれぞれ水平方向及び垂直方向に周波数及び時間の双方で分割する処理（すなわちウェーブレット変換処理）を施す。

【0015】

すなわちウェーブレット変換処理は、ある周波数帯を取り出すと共に低域周波数を2：1で間引き再びフィルタをかける操作を繰り返すことにより、再帰的に低域を分割していくようにして、高域周波数から段階的に周波数成分を抽出する処理である（以下、当該各段階をレベルと呼ぶ）。

【0016】

この結果、各コンポーネントデータD 2に基づく画像は、図2に示すように、各レベル（本実施の形態の場合5レベルまで）ごとに、水平方向が低域周波数側で垂直方向が高域周波数側の成分（以下、これをLH成分と呼ぶ）、水平方向が高域周波数側で垂直方向が低域周波数側の成分（以下、これをHL成分と呼ぶ）、水平方向が高域周波数側で垂直方向が高域周波数側の成分（以下、これをHH成分と呼ぶ）、水平方向が低域周波数側で垂直方向が低域周波数側の成分（以下、これをLL成分と呼ぶ）に分割される。

【0017】

ウェーブレット変換部5において、各レベルごとに、LH成分、HL成分、H

H成分及びL L成分に画像が分割された各コンポーネントデータD 3は、量子化部（Q：quantum）6を介して量子化された後、ビットプレーン変換部（W T B：Word to Bitplane）7に供給される。なお各コンポーネントデータD 3のうち最終階層レベル以外のL L成分は、量子化部6及びビットプレーン変換部7の各処理を行うことなく、レジスタR 4 E及びインターフェースP S Gを介してそのままS D R A M 4に記憶される。

【0 0 1 8】

ビットプレーン変換部7は、ウェーブレット変換された各コンポーネントデータD 3について、図3（A）に示すように、例えば $64 \times 64 (=4096)$ 個のワードを1個のコードブロックC Bとして、階層レベルごとにかつ帯域周波数ごとに、当該コードブロック単位に分割する。

【0 0 1 9】

さらにビットプレーン変換部7は、各コードブロックC B（図3（A））について、各画素の最上位ビット（b i t 1 5）が集まって構成するプレーンB P 1 5から最下位ビット（b i t 0）が集まって構成するプレーンB P 0までの1 6枚のビットプレーンB P 0～B P 1 5に分割する。その際、ビットプレーン変換部7は、各コンポーネントデータについての各コードブロックに対する0ビットのビットプレーン（以下、これをゼロビットプレーンと呼ぶ）を検出する。

【0 0 2 0】

この後、C P U 3は、ビットプレーン変換部7から出力された各コードブロックをレジスタR 4 Eに一旦格納しながら順次インターフェースP S Gを介してS D R A M 4に記憶させる。

【0 0 2 1】

続いてC P U 3は、S D R A M 4に記憶された各ビットプレーンを読み出しながら、インターフェースP S G及びレジスタR 5 Eを介してそれぞれ対応するビットモデル部（C B M：Coefficient Bit Modeling）8 A（8 A 0～8 A X（本実施の形態ではX=15））及び続く算術符号化部（A C：Arithmetic）9 A（9 A 0～9 A X）にパラレルに送出する。

【0 0 2 2】

その際、CPU3は、SDRAM4から読み出した各ビットプレーンのうち実際にビットが1であるビットプレーン（以下、これを処理ビットプレーンと呼ぶ）の数がコードブロック毎に相違することから、当該各コードブロックに対応するビットモデル部8Aの処理時間が変動するのを防止するように、レジスタR5Eの入出力状態を制御するようになされている。

【0023】

ここで量子化後のコードブロックCBは、図4（A）のように16枚のビットプレーンから構成され、図4（B）に示すように、最上位ビットの正負を表す1枚のビットプレーン（以下、これを符号ビットプレーンと呼ぶ）BP15と、その下位にある何枚かのゼロビットプレーンBPZと、さらに下位にある何枚かの処理ビットプレーンBPYとを有する。

【0024】

そしてCPU3は、ビットプレーン変換部7からレジスタR4Eを介して送られてくるビットプレーンを監視しながら、必要なビットプレーンのみをレジスタR5Eに送出するようにレート制御を実行させる。

【0025】

ここでビットモデル部8Aは、レジスタR5Eからパラレルで供給される各コードブロックにおける各ビットプレーンの各ビットに相当するコンテキストを作成した後、当該各ビットに同期させて対応する算術符号化部9Aに送出する。

【0026】

このビットモデル部8Aは、後段の算術符号化部9Aにおいて算術符号する順番に各ビット及び相当するコンテキストを出力する。

【0027】

ここで算術符号化部9Aでは、JPEG2000規格に準拠するエントロピー符号化をビットプレーン毎に独立して行うようになされている。従って、CPU3は、必要最小限のビットプレーンのみを符号化するために、各ビットモデル部8Aにそれぞれ供給する必要なビットプレーン数を確定させる。

【0028】

すなわち上述した図4（B）の16枚のビットプレーンのうち処理ビットプレ

ーンBPYの下位の何枚かを非処理ビットプレーンBPWとすることにより（図4（C））、残りの処理ビットプレーンBPX（ $X: Y-W$ ）のみが処理対象とされる。

【0029】

そして各算術符号化部9Aは、対応するビットモデル部8Aから送られてきたビットに対してコンテキスト毎の発生確率を演算して符号化ストリームを作成する。

【0030】

一般の算術符号化方法とは、第1に、入力ビットが「0」又は「1」である点、第2に、入力ビットを符号化するのではなくMPS（More Probable Symbol）及びLPS（Less Probable Symbol）を符号化する点、第3に、入力ビットの出現確率は入力ビットに従い学習する点、第4に、区間分割の乗算演算を加算演算によって近似する点が相違する。

【0031】

この後、CPU3は、各算術符号化部9Aからそれぞれ出力された符号化後のコードブロックをレジスタR6Eに一旦格納しながら符号化ストリームとしてインターフェースPSGを介してSDRAM4に記憶させる。

【0032】

その際、CPU3は、各算術符号化部9Aにおいて符号化処理に要する時間が相違するため、当該各算術符号化部9Aを監視し一定の符号化ストリームが作成されたら次から次へのSDRAM4に記憶させるようにレジスタR6Eを制御する。

【0033】

続いてCPU3は、SDRAM4に記憶された符号化ストリームの中でレート制御により決定された符号化ストリームのみを読み出しながら、インターフェースPSG及びレジスタR7Eを介してフォーマット生成部（FMT：format）10に送出する。

【0034】

このフォーマット生成部10は、レジスタR6Eを介してCPU3によってS

D R A M 4 から読み出された符号化ストリームについて、当該符号化ストリームがエントロピー符号化された符号列であるため、所望のストリーム順番に形成しかつ付加情報（ヘッダ）を追加して J P E G 2 0 0 0 規格に準拠したデータストリームを作成した後、外部出力する。

【 0 0 3 5 】

なお C P U 3 には、ワークメモリとしての R A M (Random Access Memory) 1 2 が接続され、必要に応じて各種データを読み出し又は書込みし得るようになされている。

【 0 0 3 6 】

(2) 符号化ビットプレーン数の確定方法

伝送レート制御により切り捨てられるビットプレーンは、結果としてビットモデル部 8 A 及び算術符号化部 9 A の各処理を行う必要がない。しかし伝送レート制御は算術符号化部 9 A において計算された発生符号量を基に制御する。そこで各ビットプレーンの係数からビットモデル部 8 A 及び算術符号化部 9 A の各処理を行わず簡易な方法で計算し、発生符号量を予測する。この方法で計算された発生符号量を予測発生符号量と呼ぶ。

【 0 0 3 7 】

この予測発生符号量に対して伝送レート制御と同じ手法で必要なビットプレーンを確定し処理ビットプレーンとする。当然予測発生符号量はビットモデル部 8 A 及び算術符号化部 9 A の各処理を行って得られた発生符号量とは異なるため確定された必要なビットプレーンに余裕をもったビットプレーンを処理ビットプレーンとする。例えば図 1 2 の灰色部分が予測発生符号量より求められた必要なビットプレーンとすると処理ビットプレーンとして各コードブロックもう 1 ビットプレーン加えて処理ビットプレーンとする。

【 0 0 3 8 】

上述したように C P U 3 は、レジスタ R 5 E を制御して各ビットモデル部 8 A に供給するコードブロックのビットプレーン数を制御することにより、当該各ビットモデル部 8 A に対応する算術符号化部 9 A における処理時間を最小限に抑える。

【0039】

その際、CPU3は、各ビットモデル部8Aにそれぞれ供給する必要なビットプレーン数を確定させる一方、必要でないビットプレーン（LSB側）を切り捨てるのであるが、以下に必要なビットプレーン数の確定方法について説明する。

【0040】

具体的には、CPU3は、図5に示すような機能ブロックで表されるアルゴリズムを有し、有意サンプル計算部3A、符号量推定部3B、1次レート制御部3C及び2次レート制御部3Dから構成される。

【0041】

まずCPU3内の有意サンプル計算部3Aでは、各コードブロックの各ビットプレーンについて、初めて“significant”（有意）になったサンプルの数を計算して出力する。

【0042】

具体的には、CPU3は、図6に示す有意サンプル計算処理手順RT1をステップSP0から開始し、続くステップSP1においてコードブロック番号cbを「0」とした後、ステップSP2に進んで、全てのサンプルsについてsignificant[s]を「0」と初期化する。

【0043】

そしてCPU3は、ステップSP3に進んで、コードブロック番号cbのビットプレーン番号bpで初めて“significant”（有意である状態、すなわち既に「1」であるビットを符号化済みであること）になったサンプルの数をCountNewSig[cb][bp]としたとき、ビットプレーン番号bpを0でかつCountNewSig[cb][bp]を0に設定した後、ステップSP4に進む。このビットプレーン番号bpはMSB側からLSB側に向けて番号が割り当てられている。

【0044】

このステップSP4において、CPU3は、コードブロック番号cb及びビットプレーン番号bpにおいて未処理のサンプルsを1つ選んだ後、ステップSP5に進んで、significant[s]が0でかつサンプルsのビットプレーン番号bp番目のビットが「1」か否かを判断する。

【0045】

このステップSP5において肯定結果が得られたとき、CPU3は、ステップSP6に進んで、significant [s] を1とし、かつCountNewSig [c b] [b p] に「1」を加えた後、ステップSP7に進む。一方ステップSP5において否定結果が得られたとき、CPU3はそのままステップSP7に進む。

【0046】

次にステップSP7において、CPU3は、未処理サンプルがあるか否かを判断し、肯定結果が得られた場合には再度ステップSP4に戻って上述と同様の処理を繰り返す一方、否定結果が得られた場合にはステップSP8に進んで、CountNewSig [c b] [b p] を後段の符号量推定部3Bに出力する。

【0047】

そしてCPU3は、ステップSP9に進んで、未処理ビットプレーンがあるか否かを判断し、肯定結果が得られた場合には、ステップSP10に進んでビットプレーン番号に「1」を加えた後、再度ステップSP3に戻って上述と同様の処理を繰り返す一方、否定結果が得られた場合には全てのビットプレーンが処理されたと判断してそのままステップSP11に進む。

【0048】

このステップSP11において、CPU3は、未処理コードブロックがあるか否かを判断し、肯定結果が得られた場合には、ステップSP12に進んでコードブロック番号に「1」を加えた後、再度ステップSP2に戻って上述と同様の処理を繰り返す一方、否定結果が得られた場合には全てのコードブロックが処理されたと判断してステップSP13に進んで当該有意サンプル計算処理手順RT1を終了する。

【0049】

このようにしてCPU3内の有意サンプル計算部3Aでは、各コードブロックにおける各ビットプレーンについて、初めて有意になったサンプルの数を計算して出力することができる。

【0050】

続いてCPU3内の符号量推定部3Bでは、各コードブロックの各ビットプレ

ーンについて、そのビットプレーンで発生する符号量の推定値を出力する。そしてCPU 3内の1次レート制御部3Cでは、ビットプレーン毎の推定符号量を入力とし、固定ビット・トランケーション（切り捨て）による制御を行い、全てのコードブロックで必要なビットプレーン数とを出力する。

【0051】

因みにCPU 3内の2次レート制御部3Dでは、後述するように、ビットプレーン毎の推定符号量のみならず、1フレーム当りの発生符号量を一定にさせるために伝送レートを制御させる量（以下、これを目標符号量と呼ぶ）をも入力として、固定ビット・トランケーションによるレート制御を行うようになされている。

【0052】

具体的には、CPU 3は、図7に示す符号量推定及び1次レート制御処理手順RT 2をステップSP 20から開始し、続くステップSP 21においてビットプレーン番号を「0」かつ利用可能なバイト数を目標バイト数に設定した後、ステップSP 22に進んで、コードブロック番号cbを「0」かつ推定符号量Bits [cb] を「0」と初期化する。

【0053】

そしてCPU 3は、ステップSP 23に進んで、ビットプレーン番号bpが「0」か否かを判断し、肯定結果が得られた場合には、ステップSP 24に進んでCountMR [cb] [bp] を「0」にした後、ステップSP 25に進む。一方ステップSP 23において、否定結果が得られた場合には、CPU 3はステップSP 26に進んでコードブロック番号cbのビットプレーン番号bpでMR（Magnitude Refinement）符号化を行う数を表すCountMR [cb] [bp] を、ビットプレーン番号が一つ前のCountMR [cb] [bp-1] とCountNewSig [cb] [bp-1] とを加算した値に設定した後、ステップSP 25に進む。なおCountMR [cb] [bp] は、 $i = 0 \sim bp - 1$ としたときの $\sum (i = 0 \sim bp - 1) \text{CountNewSig [cb] [i]}$ として定義される。

【0054】

このステップSP 25において、CPU 3は、一時的な推定符号量BitsTmpを

、RatioNewSig及びRatioMRをそれぞれパラメータとしたとき、CountNewSig [c b] [b p] × RatioNewSig + CountMR [c b] [b p] × RatioMRとして表し、利用可能なバイト数 (AvailableBytes) を 8 で割って小数部分を切り捨てるように、推定符号量をバイト数に変換すると共に、推定符号量Bits [c b] に一時的な推定符号量BitsTmpを加算する。

【 0 0 5 5 】

続いてCPU 3 は、ステップSP 2 7において、利用可能なバイト数が0 以下であるか否かを判断し、否定結果が得られた場合にはステップSP 2 8に進んで、未処理コードブロックがあるか否かを判断する一方、肯定結果が得られた場合にはそのままステップSP 2 9に進んで当該符号量推定及び1 次レート制御処理手順RT 2 を終了する。

【 0 0 5 6 】

このステップSP 2 8において肯定結果が得られた場合には、CPU 3 は、ステップSP 3 0に進んでコードブロック番号c bに「1」を加えた後、再度ステップSP 2 3に戻って上述と同様の処理を繰り返す一方、否定結果が得られた場合には全てのコードブロックが処理されたと判断してステップSP 3 1に進む。

【 0 0 5 7 】

そしてステップSP 3 1において、CPU 3 は、未処理ビットプレーンがあるか否かを判断し、肯定結果が得られた場合には、ステップSP 3 2に進んでビットプレーン番号b pに「1」を加えた後、再度ステップSP 2 2に戻って上述と同様の処理を繰り返す一方、否定結果が得られた場合には全てのビットプレーンが処理されたと判断してそのままステップSP 2 9に進んで当該符号量推定及び1 次レート制御処理手順RT 2 を終了する。

【 0 0 5 8 】

このようにしてCPU 3内の符号量推定部3 B及び1 次レート制御部3 Cでは、各コードブロックにおける各ビットプレーンについて、当該ビットプレーンで発生する推定符号量を出力すると共に、ビットプレーン毎の推定符号量とに基づいて、全てのコードブロックで必要なビットプレーン数を確定することができる。

。

【 0 0 5 9 】**(3) レート制御**

フォーマット生成部 1 0 から送出する 1 フレーム当りの発生符号量を一定にさせるために伝送レートを制御させる目標符号量を基準として、CPU 3 (図 5 で、2 次レート制御部 3 D) は、MSB 側から LSB 側の方向にかつ低域周波数側から高域周波数側の順に、輝度 Y 及び色差 C b、C r のコンポーネントの順番で、ビットプレーン単位で目標符号量になるまで選択する。

【 0 0 6 0 】

上述した符号化ビットプレーン数の確定方法では、予測発生符号量に対して必要なビットプレーンを確定したが、レート制御ではビットモデル部 8 A 及び算術符号化部 9 A の処理を行って得られた発生符号量を基に固定ビット・トラケーションによる制御を行うようになされている。

【 0 0 6 1 】

具体的には、CPU 3 は、図 1 3 に示す発生符号量予測処理手順 RT 3 をステップ SP 4 0 から開始し、続くステップ SP 4 1 において、利用可能なバイト数 (AvailableBytes) を目標バイト数からヘッダ類 (メインヘッダ (Main header)、タイルパートヘッダ (Tilepart header)、パケットヘッダ (Packet header)) の符号量を差し引いた値となるように設定する。

【 0 0 6 2 】

因みに符号化ストリーム全体として、目標符号量内に収まるように、予めヘッダ類の符号量を計算し又は見積もる必要がある。この場合、パケットヘッダの符号量は確定できないので、大まかな値を用いることを前提としている。

【 0 0 6 3 】

そして CPU 3 は、ステップ SP 4 2 に進んで、ビットプレーン番号 b p を「0」に設定した後、さらにステップ SP 4 3 に進んでコードブロック番号 c b を「0」に設定する。

【 0 0 6 4 】

続いて CPU 3 は、ステップ SP 4 4 に進んで、コードブロック番号 c b 及びビットプレーン番号 b p の実際の発生符号量のバイト数 Bytes (c b、b p) を

検出すると共に、当該バイト数Bytes (c b、b p) を利用可能なバイト数 (AvailableBytes) から差し引いた後、ステップS P 4 5に進む。

【0 0 6 5】

このステップS P 4 5において、C P U 3は、利用可能なバイト数が0以下であるか否かを判断し、否定結果が得られた場合にはステップS P 4 6に進んで、未処理コードブロックがあるか否かを判断する一方、肯定結果が得られた場合にはステップS P 4 7に進む。

【0 0 6 6】

このステップS P 4 6において肯定結果が得られた場合には、C P U 3は、ステップS P 4 8に進んでコードブロック番号c bに「1」を加えた後、再度ステップS P 4 3に戻って上述と同様の処理を繰り返す一方、否定結果が得られた場合には全てのコードブロックが処理されたと判断してステップS P 4 9に進む。

【0 0 6 7】

そしてステップS P 4 9において、C P U 3は、未処理ビットプレーンがあるか否かを判断し、肯定結果が得られた場合には、ステップS P 5 0に進んでビットプレーン番号b pに「1」を加えた後、再度ステップS P 4 2に戻って上述と同様の処理を繰り返す一方、否定結果が得られた場合には全てのビットプレーンが処理されたと判断してそのままステップS P 5 1に進んで当該発生符号量予測処理手順R T 3を終了する。

【0 0 6 8】

これに対してステップS P 4 5の肯定結果として進んだステップS P 4 7において、C P U 3は、利用可能なバイト数が0であるか否かを判断し、否定結果が得られた場合にはステップS P 5 2に進んで、コードブロック番号c bが「0」であるか否かを判断する一方、肯定結果が得られた場合にはそのままステップS P 5 1に進んで当該発生符号量予測処理手順R T 3を終了する。

【0 0 6 9】

このステップS P 5 2においてコードブロック番号c bが「0」と判断された場合には、ステップS P 5 3に進んで、コードブロック番号c bから「1」を差し引いた後、ステップS P 5 1に進んで当該発生符号量予測処理手順R T 3を終

了する。

【0070】

一方、ステップSP52においてコードブロック番号 c_b が「0」でないと判断された場合には、ステップSP54に進んで、コードブロック番号 c_b を最大値 $c_{b_{max}}$ にすると共にビットプレーン番号 b_p から「1」を差し引いた後、ステップSP51に進んで当該発生符号量予測処理手順RT3を終了する。

【0071】

ここで實際上、算術符号化部9Aにおいて計算された各ビットプレーンの発生符号量を1フレーム分蓄積した状態を図8に示す。縦軸にビットプレーン、横軸にコードブロックを割り当て、1フレームのコードブロック数を表す。

【0072】

この図2に示すコードブロック分割を例にすると、レベル3の各成分は2個、レベル2の各成分は6個、レベル1の各成分は20個の合計86個から構成される。図8において斜線部分はストリームの存在するビットプレーン、すなわち発生符号量がゼロでない有意なビットプレーンである。ビットプレーン変換部7において検出されたゼロビットプレーンは符号を発生しない。

【0073】

ここで、有意なビットプレーン（図8の斜線部分）をMSB側のビットでかつレベル数の多い方（図8の左上）から順番に発生符号量を積算する。積算し続けて目標符号量を超える場合は、目標符号量を超えない最大の積算値のビットプレーンで終わる。すなわち図9において、縦軸がビット3（ $bit\ 3$ ）で横軸がレベル1（Level）のHL成分の途中で終わる。積算し続けて目標符号量を超えない場合は、全てのビットプレーンについて行う。以上積算されたビットプレーンを必要なビットプレーン（図9の斜線部分）と確定する。

【0074】

実際には各ウェーブレットの階層レベルと帯域に重みをつける。すなわちウェーブレット変換部5（図1）より得られた係数を量子化部6において量子化する。

【0075】

図10の例においては、レベル3のLL成分は1で割り（量子化しない）、レベル3のHL成分及びLH成分は2で割り、レベル3のHH成分は4で割るような量子化を行う。またレベル2のHL成分及びLH成分は16で割り、レベル2のHH成分は32で割るような量子化を行う。さらにレベル1のHL成分及びLH成分は128で割り、レベル1のHH成分は256で割るような量子化を行う。

【0076】

このように図8に対応する図11に示すような量子化されたビットプレーンに対して伝送レートの制御を行い、図12に示すように、必要なビットプレーンを確定する。

【0077】

このようにしてCPU3では、各コードブロックにおける各ビットプレーンについて、MSB側からLSB側の方向にかつ低域周波数側から高域周波数側の順に、輝度Y及び色差Cb、Crのコンポーネントの順番で、ビットプレーン単位で目標符号量になるまで選択することにより、最終的に固定ビット・トランケーションによるレート制御を行うことができる。

【0078】

（4）本実施の形態による動作及び効果

以上の構成において、この符号化装置1では、ウェーブレット変換処理された各コンポーネントデータD3について、所定サイズのワードをコードブロックとして、階層レベルごとにかつ帯域周波数ごとに分割した後、さらに当該各コードブロックを、各画素の最上位ビットから最下位ビットまでの複数のビットプレーンに分割する。

【0079】

その際、複数のビットプレーンのうち、MSB側のビットでかつレベル数の多い方の有意な状態にあるビットプレーンを残す一方、細かい情報が割り当てられた下位（LSB側）のビットプレーンを切り取るように、SDRAM4から読み出してレジスタR5Eに供給するコードブロック単位のデータをレート制御する。

【0080】

この結果、各ビットモデル部 8 A は、レジスタ R 5 E からパラレルで供給される各コードブロックにおける各ビットプレーンの各ビットに相当するコンテキストを作成した後、対応する各算術符号化部 9 A において、当該コンテキストを各ビットに同期させて順番に J P E G 2 0 0 0 規格に準拠するエントロピー符号化処理を行う際に、必要最小限の時間で処理させることができる。

【0081】

そして各算術符号化部 9 A は、対応するビットモデル部 8 A から送られてきたビットに対してコンテキスト毎の発生確率を演算した後、各算術符号化部 9 A の演算結果に基づいて、各コードブロックにおける各ビットプレーンについて、M S B 側から L S B 側の方向にかつ低域周波数側から高域周波数側の順に、輝度 Y 及び色差 C b、C r のコンポーネントの順番で、ビットプレーン単位で目標符号量になるまで選択するようにして、1 フレーム当りの発生符号量を一定にさせたストリームを S D R A M 4 を介してフォーマット生成部 10 に送る。フォーマット生成部 10 は、所望のストリーム順番に並べ替えると共にヘッダ類を付加して符号化ストリームを作成する。

【0082】

以上の構成によれば、この符号化装置 1 では、J P E G 2 0 0 0 規格に準拠する符号化処理を行う際に、ユーザから見て比較的細かい情報である部分を取り除くようにして、1 フレーム当りの発生符号量を一定になるようにデータの伝送レートを制御することにより、符号化処理を最小限の時間に抑えることができ、かくして符号化処理に要する伝送効率を向上することができる。さらにビットモデル部 8 A と算術符号化部 9 A の前に発生符号量を予測することにより、ビットモデル部 8 A 及び算術符号化部 9 A の各処理を最小限に抑えることができる。

【0083】

(5) 他の実施の形態

なお上述のように本実施の形態においては、本発明の符号化装置として図 1 に示す構成のものを適用するようにした場合について述べたが、本発明はこれに限らず、ユーザから見て比較的細かい情報である部分に相当する所定数分のビットプレーンを取り除いて符号化処理に要する伝送効率を向上し、符号化処理を最小

限の時間に抑えられることができれば、この他種々の構成のものに広く適用することができる。

【0084】

また本実施の形態においては、フレーム単位のウェーブレット変換を施したウェーブレット係数で所定サイズのコードブロック（フィルタリング係数）に分割するフィルタリング生成手段として、図1のウェーブレット変換部5を適用するようにして、画像データを高域周波数側から再帰的かつ階層的に低域周波数成分を抽出した後、階層レベルごとにかつ帯域周波数ごとに、各コードブロックに分割するようにした場合について述べたが、本発明はこれに限らず、入力された画像データをフィルタリング処理を施すことによりフィルタリング係数を生成することができれば、この他種々の構成のものに広く適用することができる。

【0085】

さらに本実施の形態においては、フィルタリング生成手段により分割されたコードブロック（フィルタリング係数）を、各画素の最上位ビットから最下位ビットまでの複数のビットプレーンに分割する分割手段として、図1のビットプレーン変換部7を適用するようにした場合について述べたが、本発明はこれに限らず、この他種々の構成のものに広く適用することができる。

【0086】

さらに本実施の形態においては、ビットプレーン変換部（分割手段）7により分割された複数のビットプレーンをSDRAM4（図1）に記憶するようにした場合について述べたが、本発明はこれに限らず、この他種々の構成からなる記憶手段に記憶させるようにしても良い。

【0087】

さらに本実施の形態においては、SDRAM4に記憶されている複数のビットプレーンのうち、下位側から所定数のビットプレーンを取り除いた後、残りの各ビットプレーンのみを読み出して並列的に出力する読出制御手段として、CPU3を適用するようにした場合について述べたが、本発明はこれに限らず、この他種々の構成のものに広く適用することができる。

【0088】

【発明の効果】

上述のように本発明によれば、符号化装置において、入力された画像データをフィルタリング処理を施すことによりフィルタリング係数を生成するフィルタリング生成手段と、フィルタリング係数を各画素の最上位ビットから最下位ビットまでの複数のビットプレーンに分割する分割手段と、複数のビットプレーンのうち、下位側から所定数のビットプレーンを取り除いた後、残りの各ビットプレーンのみを読み出して並列的に出力する読出制御手段と、読出制御手段により並列的に出力された複数のビットプレーンについてそれぞれ符号化処理する複数の符号化手段とを設け、各符号化手段が符号化処理を行う際に 1 フレーム当りの発生符号量を一定にさせるように、読出制御手段は、取り除くビットプレーンの数を決定するようにしたことにより、ユーザから見て比較的細かい情報である部分に相当する所定数分のビットプレーンを取り除いた分だけ、各符号化手段がそれぞれ符号化処理を行う前段階での処理時間の変動を未然に防止することができ、かくして符号化処理に要する伝送効率を格段と向上し得る符号化装置を実現できる。

【0 0 8 9】

また本発明によれば、符号化方法において、入力された画像データをフィルタリング処理を施すことによりフィルタリング係数を生成する第 1 のステップと、フィルタリング係数を各画素の最上位ビットから最下位ビットまでの複数のビットプレーンに分割する第 2 のステップと、分割された複数のビットプレーンのうち、下位側から所定数のビットプレーンを取り除いた後、残りの複数のビットプレーンのみを読み出して並列的に出力する第 3 のステップと、並列的に出力された複数のビットプレーンについてそれぞれ符号化処理する第 4 のステップとを設け、第 4 のステップにおいて各符号化処理を行う際に 1 フレーム当りの発生符号量を一定にさせるように、第 3 のステップでは、取り除くビットプレーンの数を決定するようにしたことにより、ユーザから見て比較的細かい情報である部分に相当する所定数分のビットプレーンを取り除いた分だけ、各符号化処理を行う前段階での処理時間の変動を未然に防止することができ、かくして符号化処理に要する伝送効率を格段と向上し得る符号化方法を実現できる。

【図面の簡単な説明】**【図 1】**

本実施の形態による符号化装置の構成を示すブロック図である。

【図 2】

ウェーブレット変換処理の説明に供する略線的な平面図である。

【図 3】

コードブロック及びビットレーンの説明に供する略線図である。

【図 4】

量子化後のコードブロック及びこれに対応するビットプレーンの内容の説明に供する略線図である。

【図 5】

C P U の機能ブロックの説明に供する略線図である。

【図 6】

有意サンプル計算処理手順の説明に供するフローチャートである。

【図 7】

符号量推定及び 1 次レート制御処理手順の説明に供するフローチャートである。

。

【図 8】

1 フレーム分蓄積された各ビットプレーンの発生符号量の説明に供する平面図である。

【図 9】

1 フレーム分蓄積された各ビットプレーンの発生符号量の説明に供する平面図である。

【図 1 0】

1 フレーム分蓄積された各ビットプレーンの発生符号量の説明に供する平面図である。

【図 1 1】

1 フレーム分蓄積された各ビットプレーンの発生符号量の説明に供する平面図である。

【図 1 2】

1 フレーム分蓄積された各ビットプレーンの発生符号量の説明に供する平面図である。

【図 1 3】

発生符号量予測処理手順の説明に供するフローチャートである。

【符号の説明】

1 ……符号化装置、2 ……ビデオ入力部、3 ……C P U、4 ……S D R A M、
5 ……ウェーブレット変換部、6 ……量子化部、7 ……ビットプレーン変換部、
8 A ……ビットモデル部、9 A ……算術符号化部、1 0 ……フォーマット生成部、
1 1 ……レート制御部、P S G ……インターフェース、R 1 E ~ R 7 E ……レジスタ、
R T 1 ……有意サンプル計算処理手順、R T 2 ……符号量推定及び 1 次
レート制御処理手順、R T 3 ……発生符号量予測処理手順。

【書類名】 図面

【図 1】

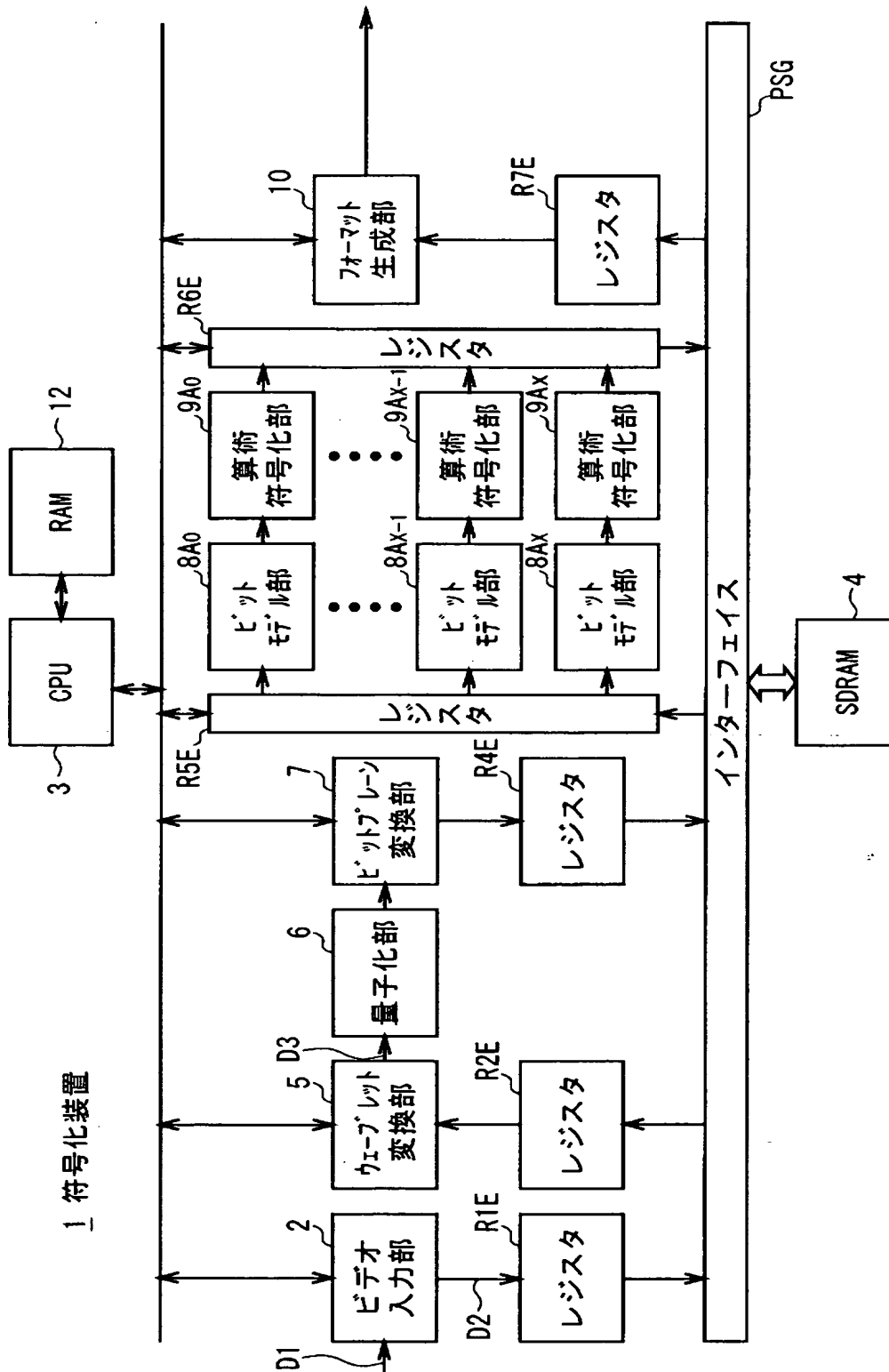


図 1 本実施の形態による符号化装置の構成

【図 2】

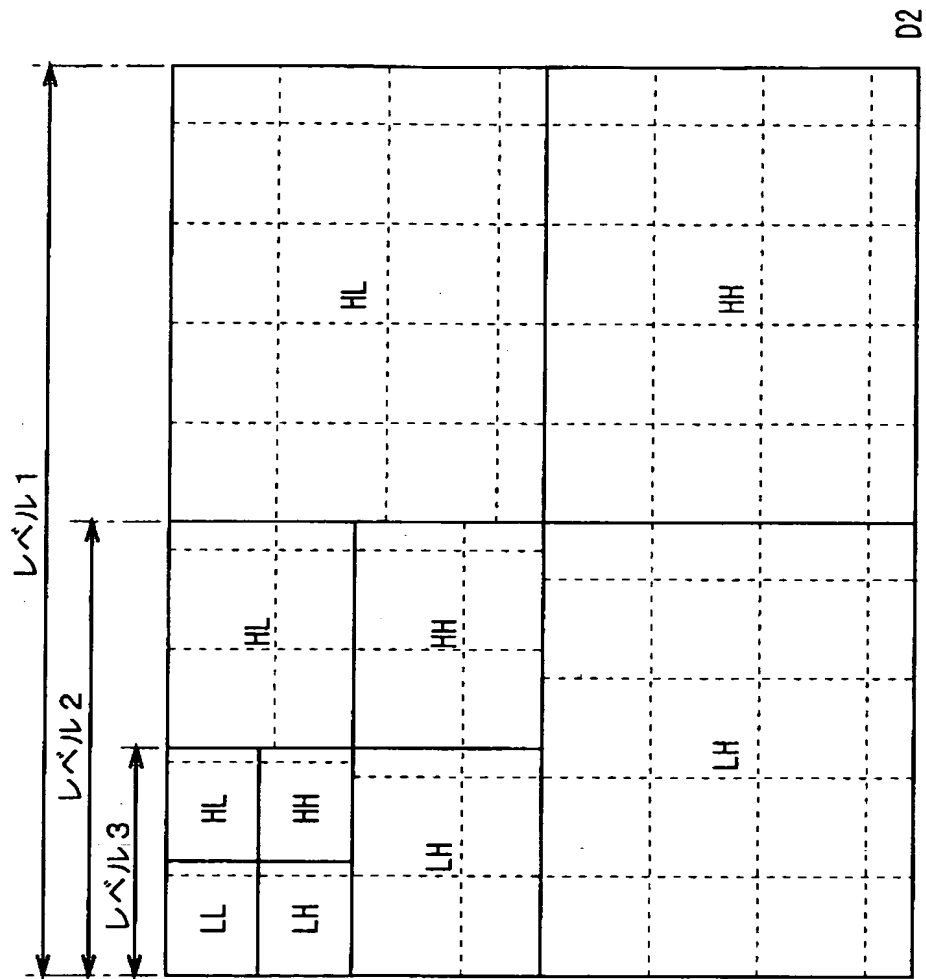
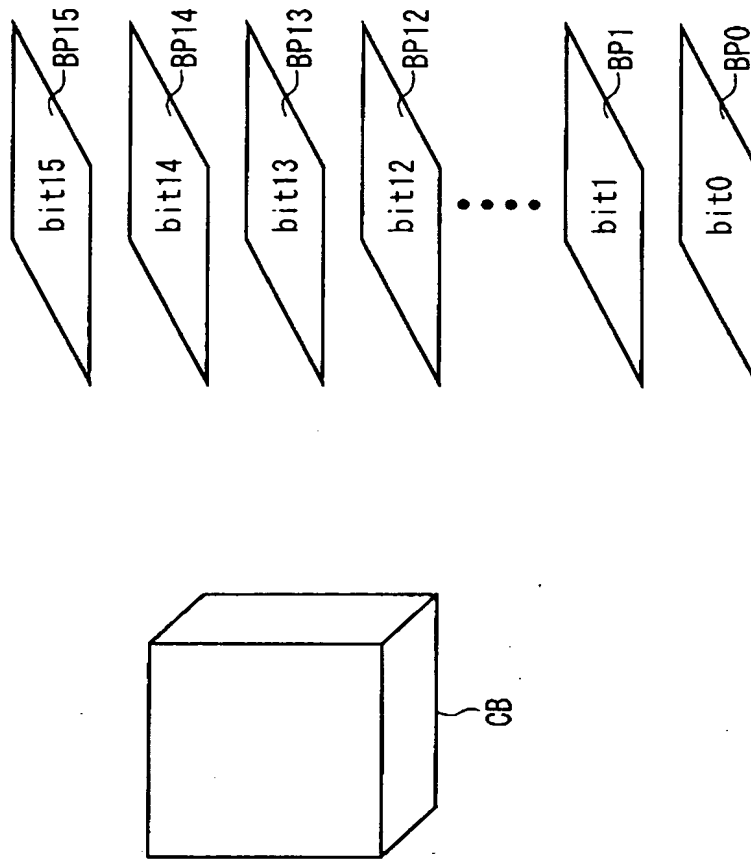


図 2 ウェーブレット変換処理

【図 3】



(A) (B)

図 3 コードブロック及びビットレーン

【図 4】

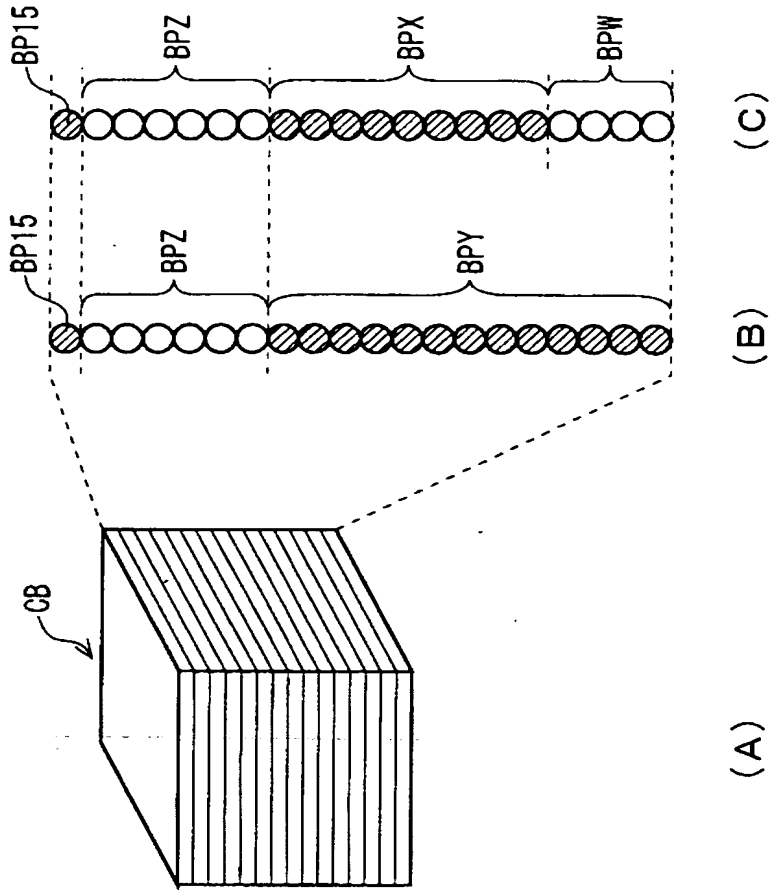


図 4 量子化後のコードブロックとこれに対応するビットプレーンの内容

【図5】

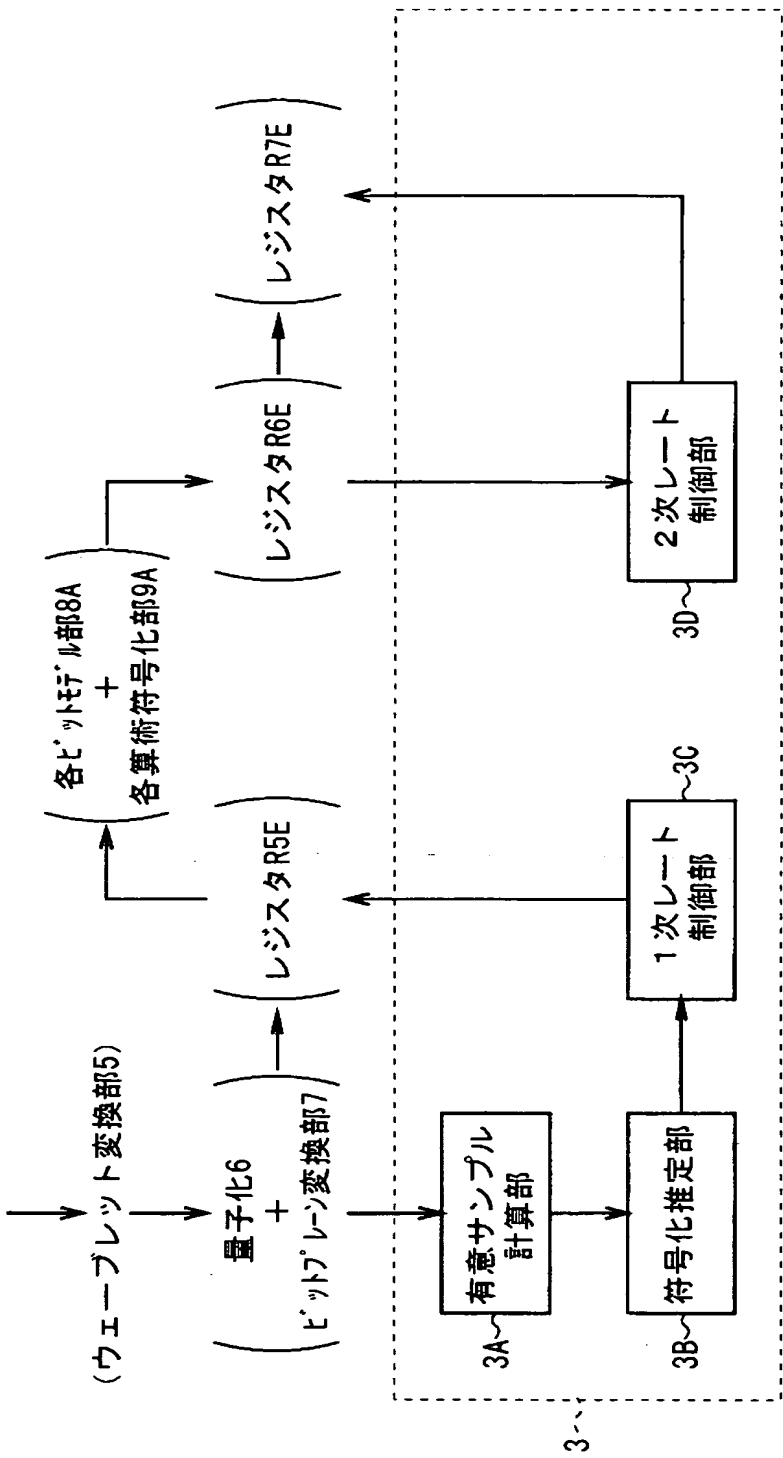


図5 CPUの機能ブロックの概略構成

【図 6】

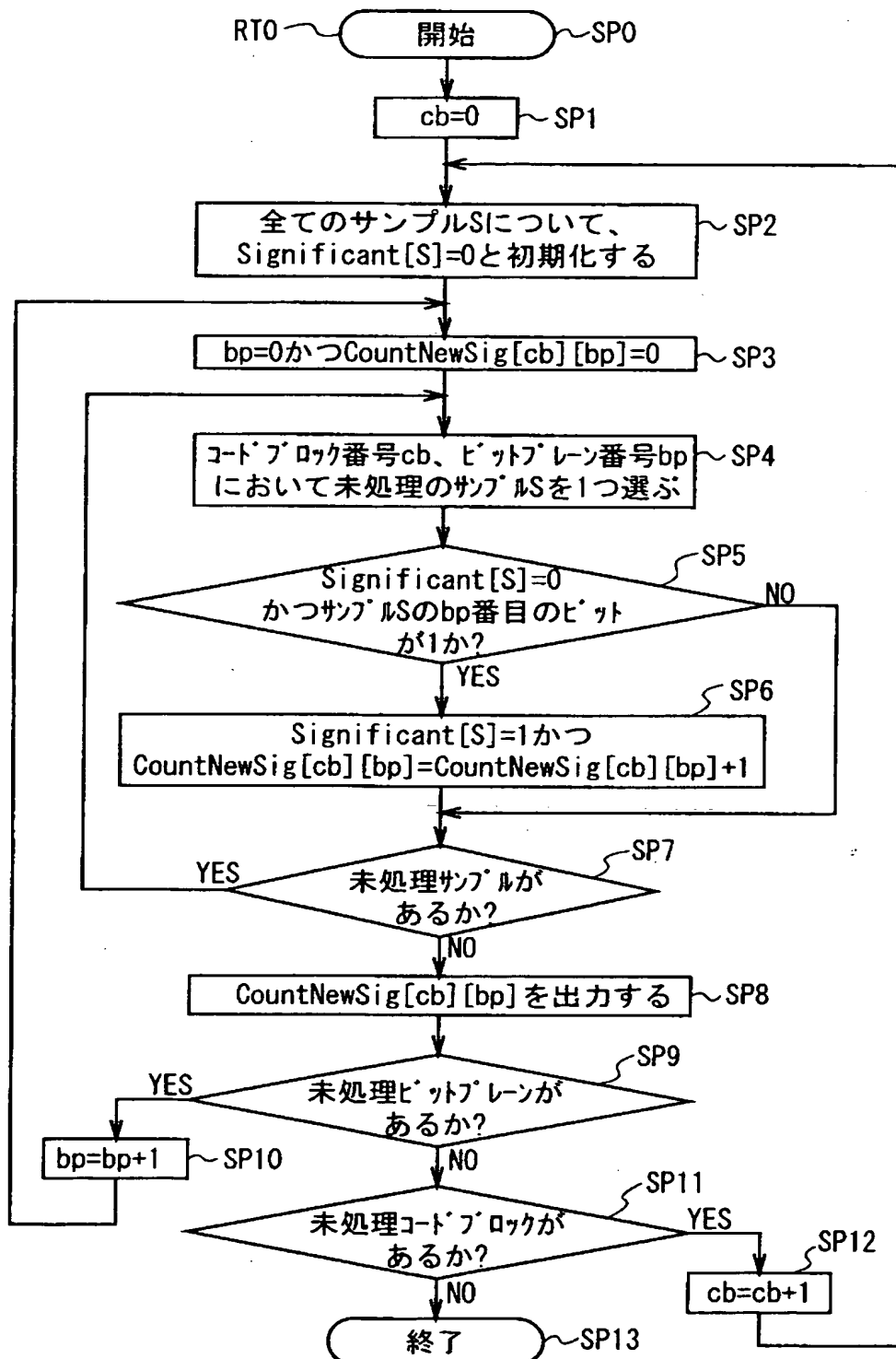


図 6 有意サンプル計算処理手順

【図 7】

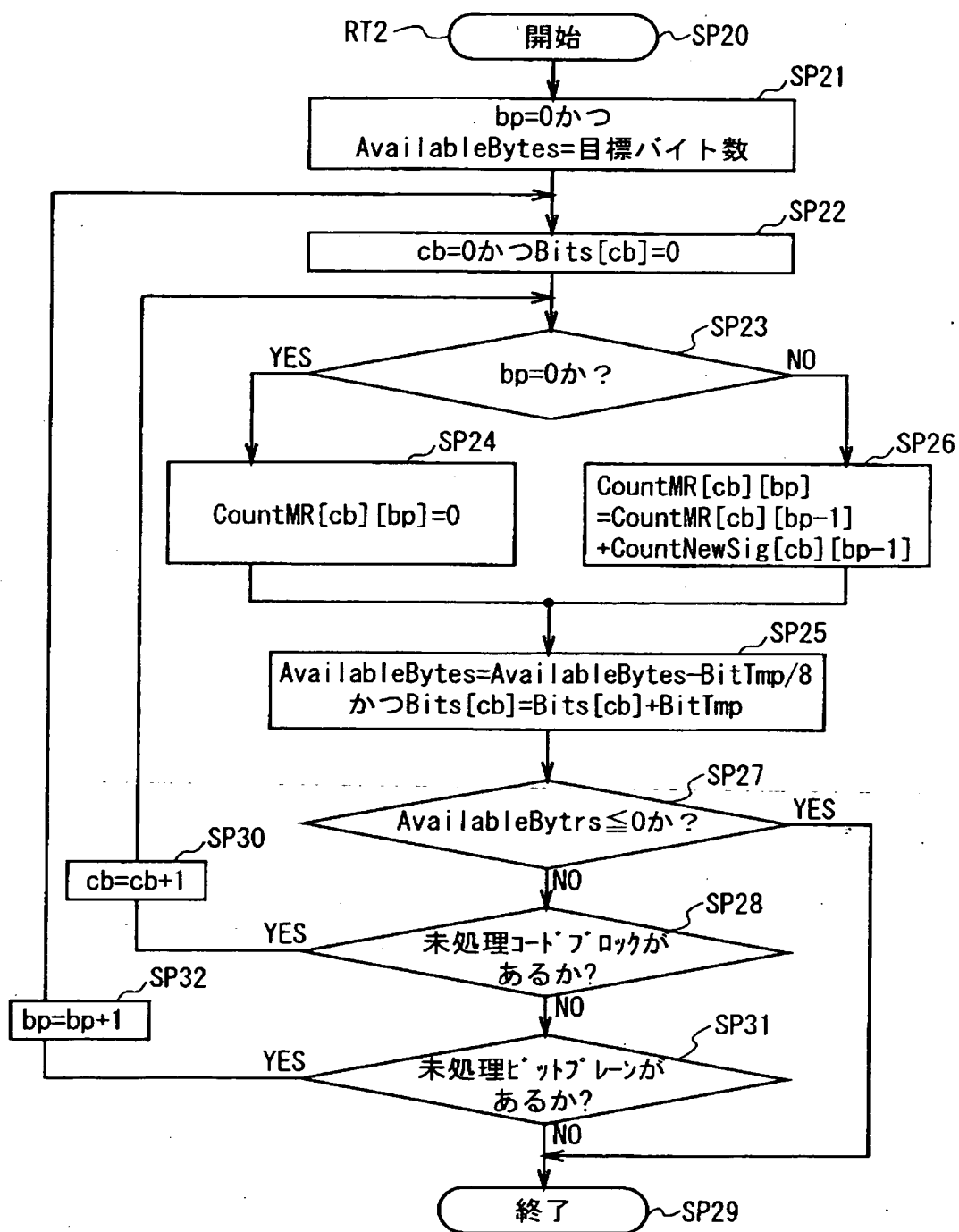


図 7 符号量推定及び 1 次レート制御処理手順

【図 8】

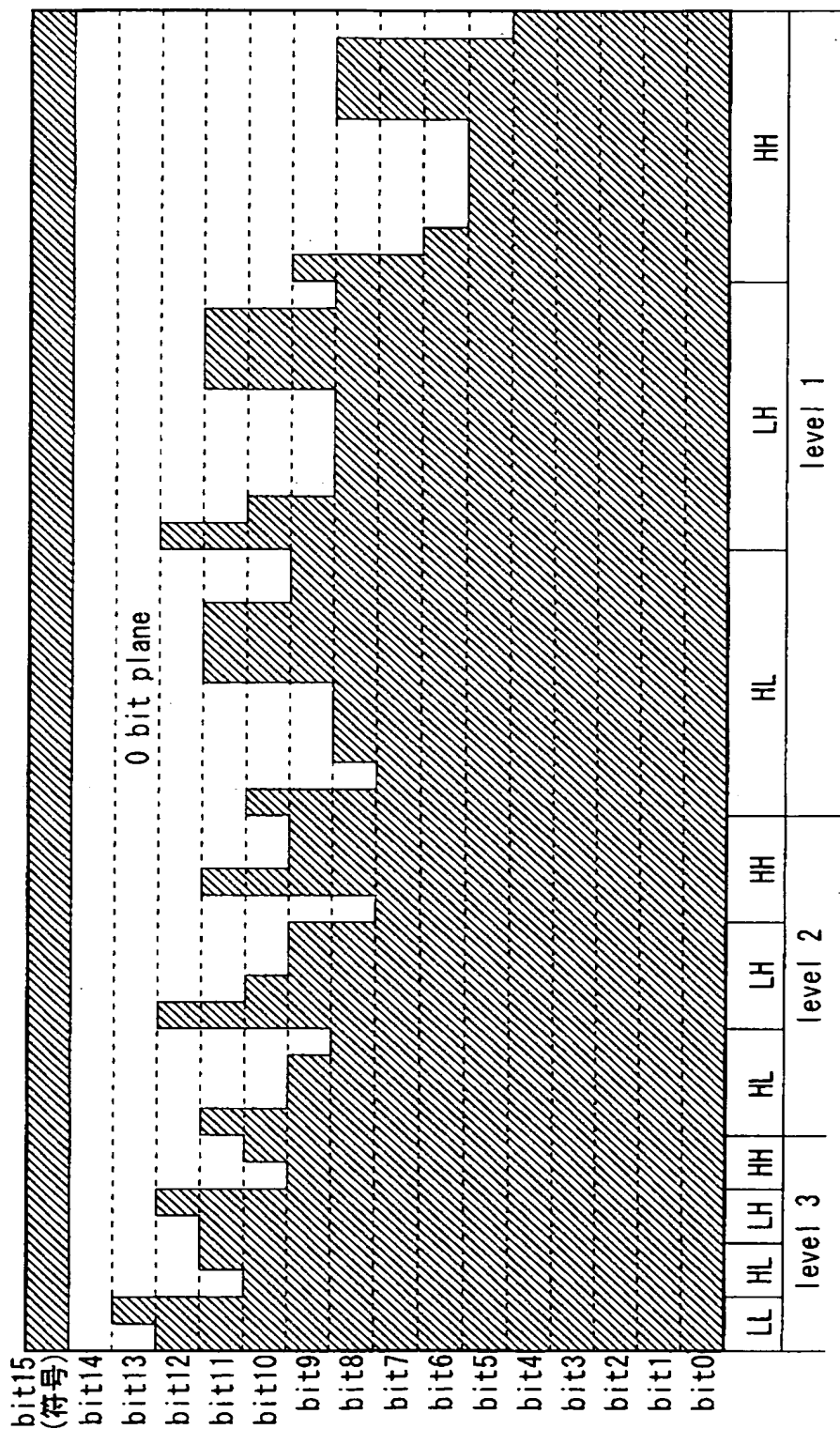


図 8 1フレーム分蓄積された各ビットプレーンの発生符号量 (1)

【図 9】

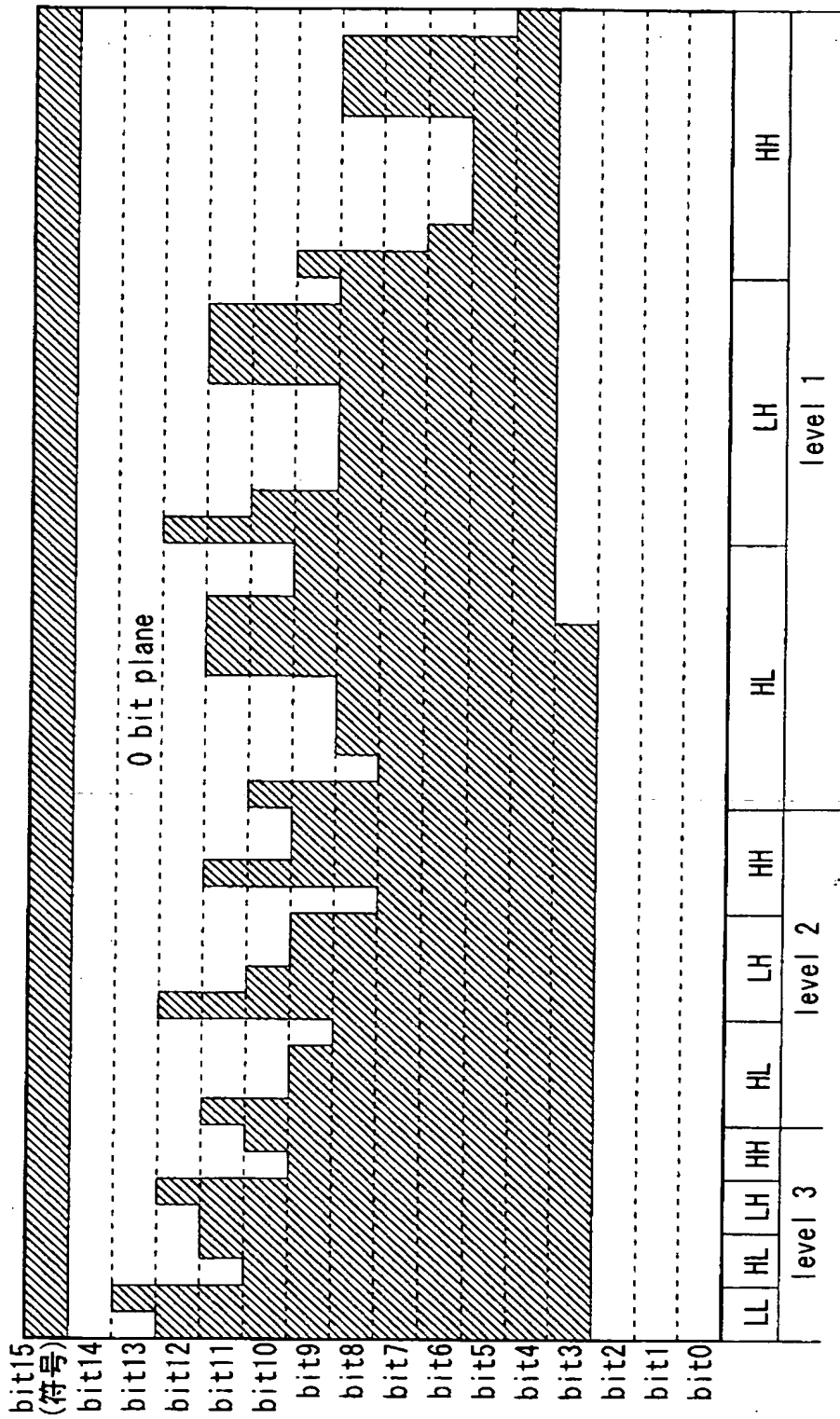


図 9 1フレーム分蓄積された各ビットプレーンの発生符号量 (2)

【図 10】

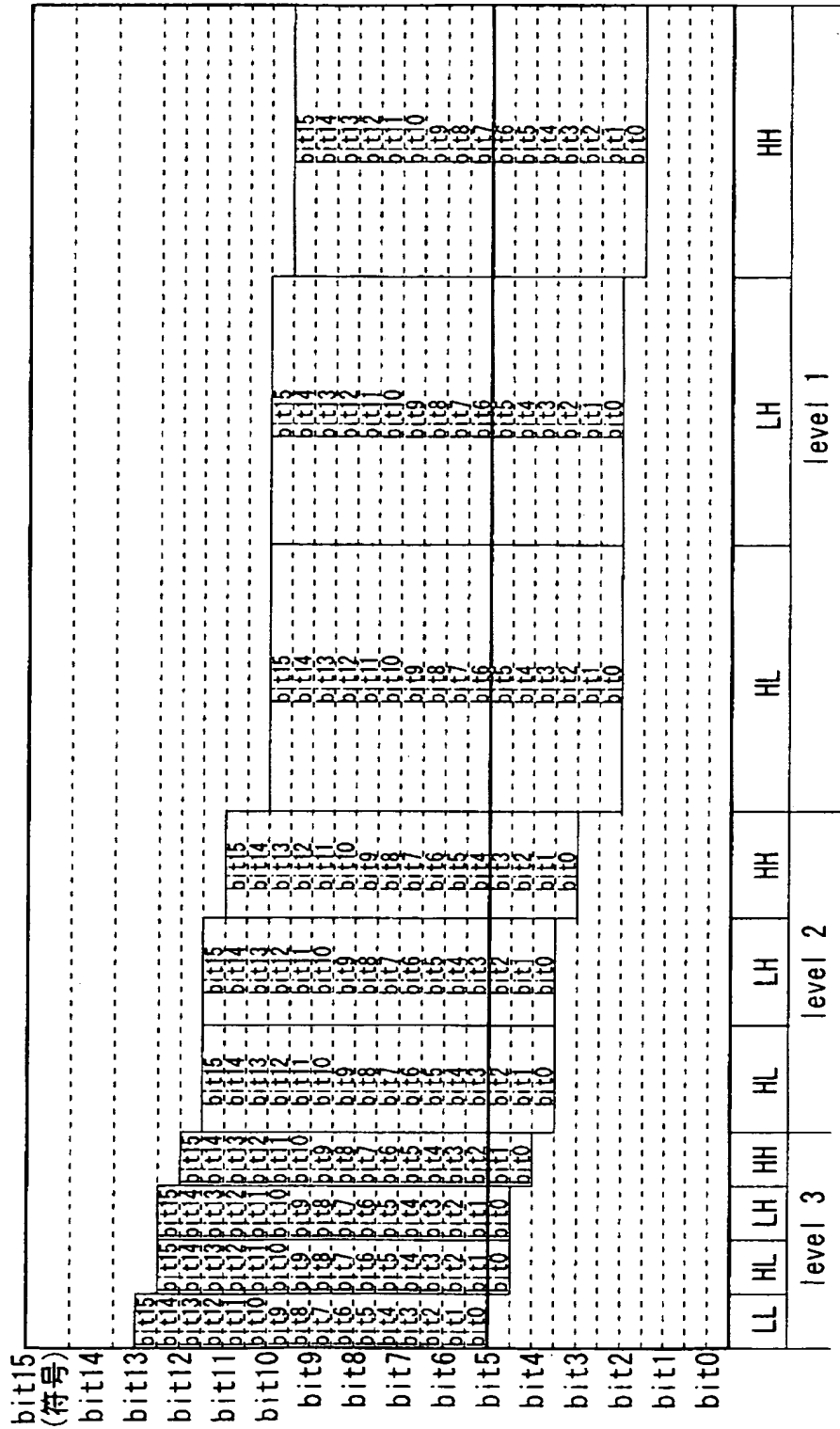


図 10 1 フレーム分蓄積された各ビットプレーンの発生符号量 (3)

【図 11】

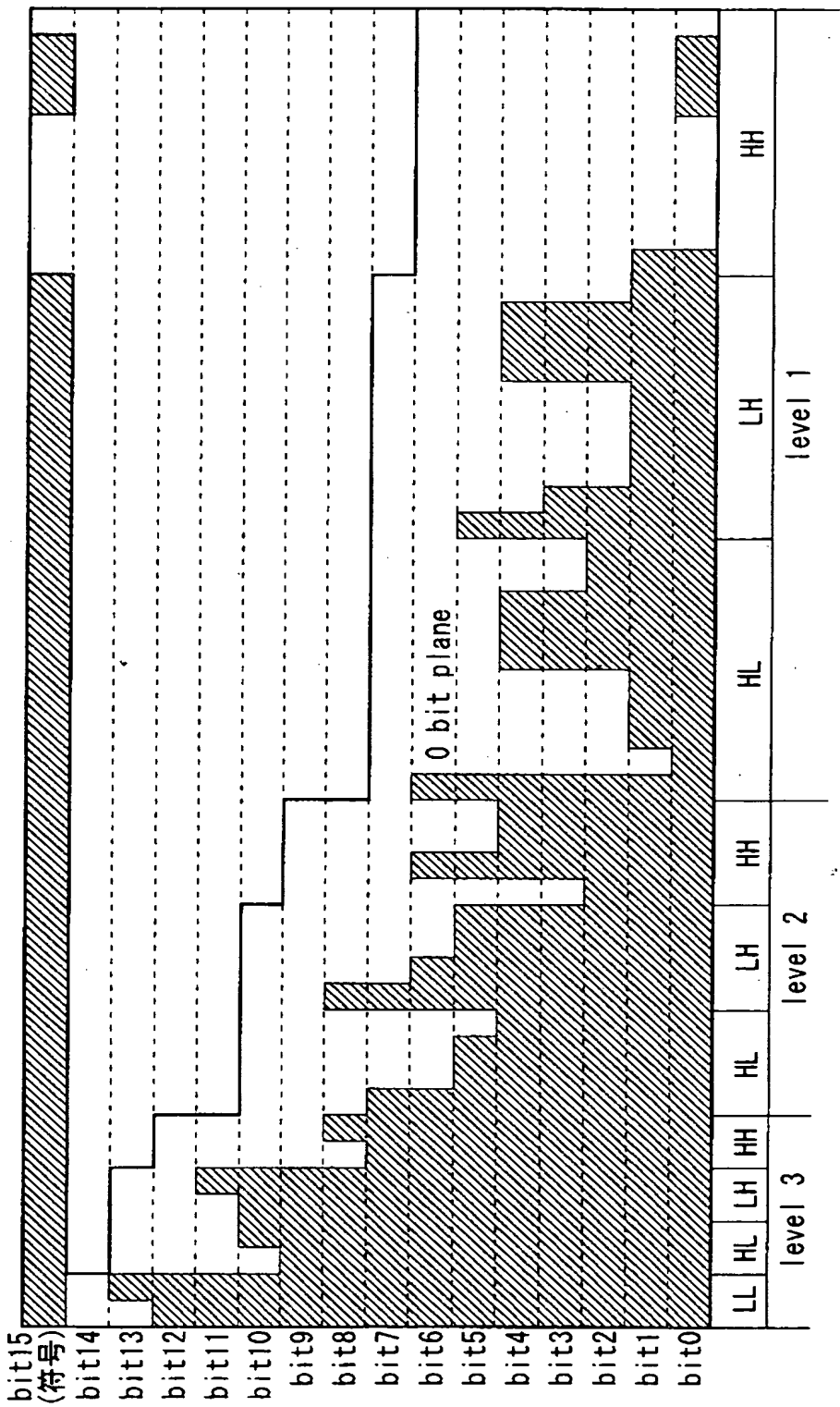


図 11 1フレーム分蓄積された各ビットプレーンの発生符号量 (4)

【図 12】

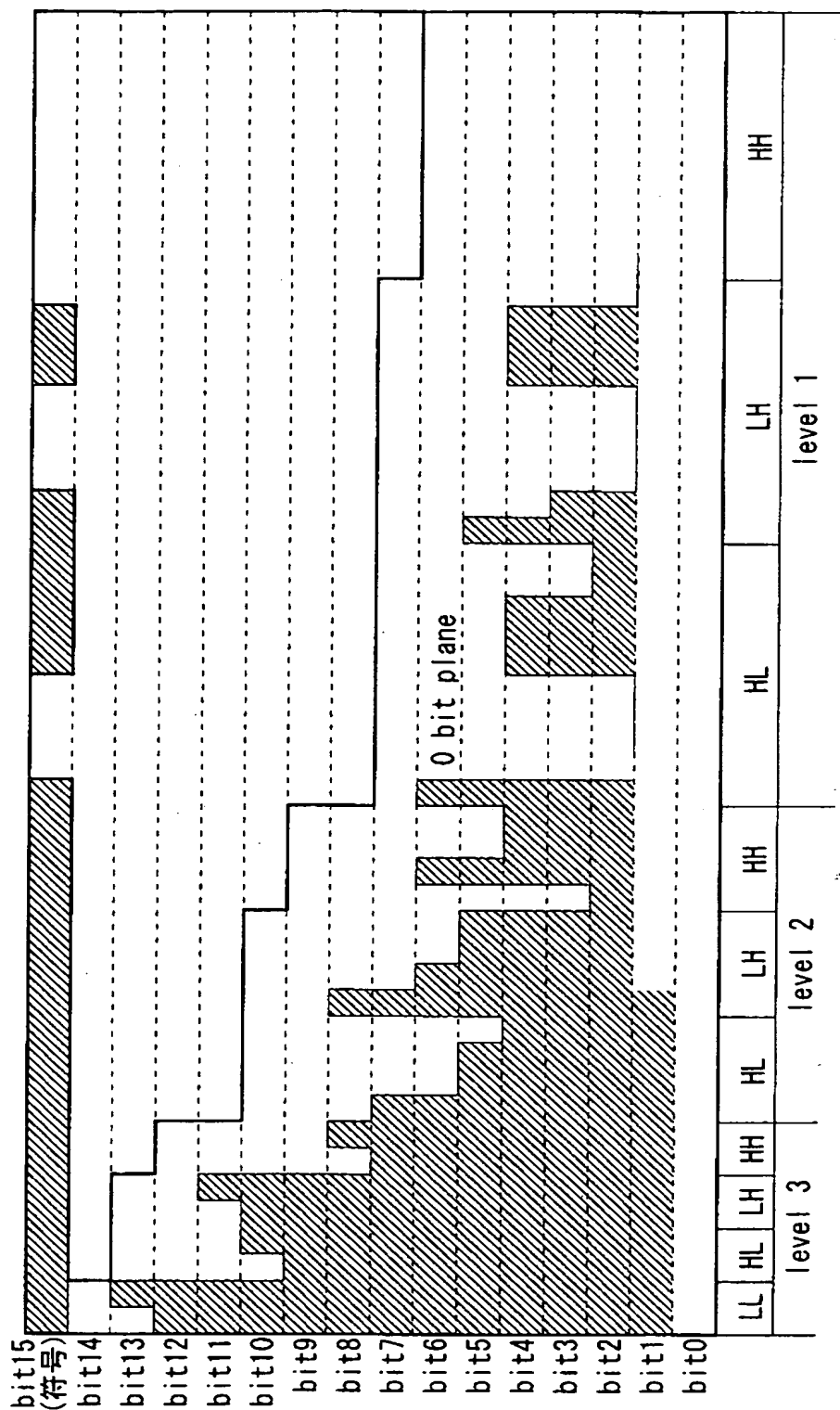


図 12 1 フレーム分蓄積された各ビットプレーンの発生符号量 (5)

【図 13】

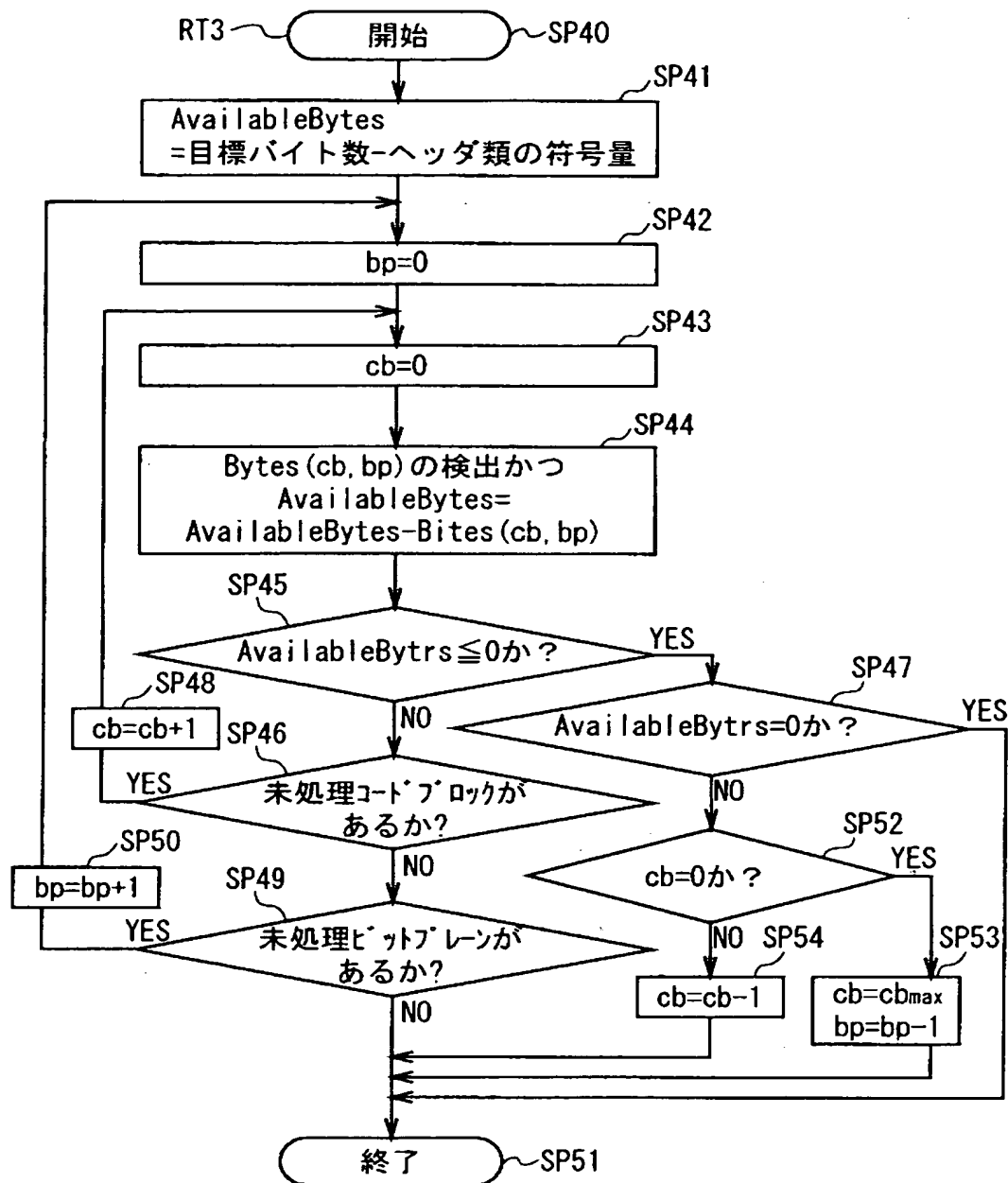


図 13 発生符号量予測処理手順

【書類名】 要約書

【要約】

【課題】

本発明は、符号化処理に要する伝送効率を格段と向上し得る符号化装置及び符号化方法を実現するものである。

【解決手段】

各符号化手段が符号化処理を行う際に1フレーム当りの発生符号量を一定にさせるように、読出制御手段は、取り除くビットプレーン数を決定するようにしたことにより、ユーザから見て比較的細かい情報である部分に相当する所定数分のビットプレーンを取り除いた分だけ、各符号化手段がそれぞれ符号化処理を行う前段階での処理時間の変動を未然に防止することができるようにした。

【選択図】 図9

特願 2 0 0 3 - 1 0 9 0 5 3

出 願 人 履 歴 情 報

識別番号 [0 0 0 0 0 2 1 8 5]

1. 変更年月日	1 9 9 0 年 8 月 3 0 日
[変更理由]	新規登録
住 所	東京都品川区北品川 6 丁目 7 番 3 5 号
氏 名	ソニー株式会社